UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/769,231	01/30/2004	Cary L. Bates	ROC920030244US1	9641

7590 07/10/2006

Grant A. Johnson
IBM Corporation, Dept. 917
3605 Highway 52 North
Rochester, MN 55901-7829

EXAMINER	
NGUYEN, PHILLIP H	
ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 07/10/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/769,231	BATES ET AL.	
	Examiner	Art Unit	
	Phillip H. Nguyen	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *30 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the original filing of January 30, 2004. Claims 1-37 are pending and have been considered below.

Specification

2. The disclosure is objected to because of the following informalities: In paragraph [0020] and claim number 24, applicant is required to defined the term XML. For the examining purposes, the examiner interprets it as Extensible Markup Language.

Appropriate correction is required.

Claim Objections

3. Claim 12 is objected to because of the following informalities: Claim 12 should depend on claim 11, not claim 9.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-18, 21-37 are rejected under 35 U.S.C. 102(b) as being anticipated by Bates (US 6077312).

6. Claim 1: Bates discloses a computer implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information, method comprising:

a. At a halted position in the code ([0006; 0043]);

b. Accessing, via the debugger, a debug history repository (Calling Stack) comprising a plurality of history records each containing data describing code state information for a previously encountered given state of the code ([0014-0015]);

c. Determining whether a current state of the code matches a previously encountered given state described in a history record in the debug history repository (Calling Stack) ([0018]); and

d. Providing an indication of the match to a user via the one or more interfaces whereby the user is allowed to view debug information contained in the history record corresponding to the match, the debug information describing at least an aspect of the previously encountered given state ([0047-0049]).

Claim 2: Bates discloses the method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 1 above, and further disclose the given state of the code is one at which the code was halted following execution ([0043]).

Claim 3: Bates discloses the method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 1 above, and further disclose providing the user the indicator of the match comprises displaying a selectable graphical element which, when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match ([0047]).

Claim 4: Bates discloses the method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 1 above, and further disclose the debugger and debug history repository are part of a distributed environment (Fig 2, item 40).

Claim 5: Bates discloses the method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 1 above, and further disclose the code is from the group consisting of procedural program code, object oriented program code, and combinations thereof ([0011-0012]).

Claim 6: Bates discloses the method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 1 above, and further disclose the given state of the code is defined at least in part by a debugger control point encountered during execution of the code and having caused the code to be stopped by the debugger at the halted position ([0024-0027]).

Claim 7: Bates discloses the method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 1 above, and further disclose the given state of the code is defined at least in part by an evaluation of a variable, the evaluation having been performed at the request of the user at the halted position ([0026-0027]).

Claim 8: Bates discloses the method as in claim 7 above; and further discloses providing the user the indication of the match comprises displaying a series of previously performed variable evaluations performed during a previous stoppage of the code at the halted position ([0047]).

Claim 9: Bates discloses a computer implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information, the method comprising:

a. At a halted position in the code ([0043]);

b. Accessing, via the debugger, a debug history repository (Calling Stack) comprising a plurality of history records each containing data describing code state information for a given state of the code while stoppage at a halted position and data describing a series of variable evaluations performed during a stoppage at a position in the code ([0014-0015]);

c. Performing at least one evaluation of one or more variables in the code ([0027]; Fig 3, item 58); and

d. For each evaluation, determining whether a current state of the code matches a given state described in a history record in the debug history repository; and if so, providing a user an indication of the match via the one or more interfaces ([0047]).

Claim 10: Bates discloses a computer implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 9 above, and further discloses providing the user the indication of the match comprises displaying a selectable graphical element which, when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match ([0047]).

Claim 11: Bates discloses a computer implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 9

above, and further discloses providing the user the indication of the match comprises displaying the data describing the series of variable evaluations contained in the history record corresponding to the match ([0047]).

Claim 12: Bates discloses a computer implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 11 above, and further discloses providing the user the indication of the match comprises displaying a selectable graphical element which, when selected by the user through an input device, reveals the debug information contained in the history record corresponding to the match ([0047]).

Claim 13: Bates discloses the method for debugging code as in claim 12 above, and further discloses the debug history information comprises user commentary (Fig. 6, item 130).

Claim 14: Bates discloses the method for debugging code as in claim 12 above, and further discloses the debug information comprises user commentary describing operations performed during a previous stoppage of the code while in a state matching the current state (Fig. 6, item 130).

Claim 15: Bates discloses a computer implemented method for debugging code using a debugger comprising one or more interfaces for specifying debugger operations and displaying debug information as in claim 9

above, and further discloses after performing the at least one evaluation, displaying a user interface configured to allow the user to create a history record for the at least one evaluation ([0051]).

Claim 16: Bates discloses the method as in claim 15 above, and further discloses the user interface is configured to allow the user to specify a state to be associated with the history record to be created ([Fig 6, item 130]).

Claim 17: Bates discloses the method as in claim 15 above, and further discloses the at least one evaluation comprises a plurality of evaluations and wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate one of the plurality of evaluations as a primary state for the history record to be created, wherein the primary state is used to determine a match between the history record to be created and a subsequent state at a subsequent halted position in the code ([0047]).

Claim 18: Bates discloses a computer readable medium containing a debugger program which, when executed, performs an operation to facilitate debugging of code, the operation comprising:

a. Collecting state information related to the current stopped position ([0043]);

b. Querying a debug history to determine whether a current state of the code matches a given state described in a history record in the debug history repository ([0014-0015]); and

c. If so, providing a user an indication of the match via the one or more interfaces ([0047]).

Claim 21: Bates discloses a computer readable medium containing a debugger program which, when executed, performs an operation to facilitate debugging of code as in claim 18 above, and further discloses the state information is defined at the least in part by the evaluation of one or more variables, the evaluation having been performed at the request of the user at the halted position ([0024-0026]).

Claim 22: Bates discloses a computer readable medium containing a debugger program which, when executed, performs an operation to facilitate debugging of code as in claim 18 above, and further discloses the debug history repository is queried using filters selected from the group consisting of developer name, time, and combinations thereof ([0052]).

Claim 23: Bates discloses a computer system for debugging code, comprising:

a. A debug history repository comprising a plurality of history records each containing data describing code state information for a given state of the code under debug ([0014-0015]);

b. A debugger comprising one or more interfaces for specifying debugger operations and displaying debug information; wherein the debugger is configured to create the plurality of history records in the debug history repository; access the plurality of history records in the debug history repository; determine at a given halted position in the code, whether a current state of the code matches a given state described in a history record of the plurality of history records ([0043-0047]); and

c. If a matching state is determined, provide a user an indication of the match via the one or more interfaces whereby the user is allowed to view debug information contained in the history record corresponding to the match ([0047]).

Claim 24: Bates discloses a computer system for debugging code as in claim 23 above, and further discloses the repository is a database selected from the group consisting of a relational database, object-relational database, XML database, and combinations thereof ([0011-0012]).

Claim 25: Bates disclose a computer system for debugging code as in claim 23 above, and further discloses the debugger queries the debug history repository based on available current state information ([0018]).

Claim 26: Bates discloses a computer system for debugging code as in claim 23 above, and further discloses the one or more interfaces comprise a graphical user interface (Fig. 6).

Claim 27: Bates discloses a computer system for debugging code as in claim 23 above, and further discloses debugger is configured to provide the user the indication of the match by displaying a selectable graphical element which, when selected by the user through an input device, reveals debug information contained in the history record corresponding to the match ([0047]).

Claim 28: Bates discloses a computer system for debugging code as in claim 27 above, and further discloses the debug history information comprises user commentary (Fig. 6, item 130).

Claim 29: Bates discloses a computer system for debugging code as in claim 27 above, and further discloses the debug information comprises user commentary describing debugging operations performed during a previous stoppage of the code while in a state matching the current state (Fig. 6, item 130).

Claim 30: Bates discloses a computer system for debugging code as in claim 23 above, and further discloses each history record further contains data

describing a series of variable evaluations performed during a stoppage at a position in the code ([0043-0047]).

Claim 31: Bates discloses a computer system for debugging code as in claim 30 above, and further discloses the debugger is further configured to perform at least one evaluation of one or more variables in the code at the halted position; and wherein the debugger performs the determination of whether the current state of the code matches a given state described in a history record for each evaluation; and wherein, if the matching state is determined and the indication of the match provided, the user is allowed to view the data contained in the history record corresponding to the match and describing the series of variable evaluations performed during the stoppage at a position in the code ([0043-0047]).

Claim 32: Bates discloses a computer system for debugging code as in claim 30 above, and further discloses the history record corresponding to the match further comprises user commentary (Fig 6, item 130).

Claim 33: Bates discloses a computer system for debugging code as in claim 30 above, and further discloses the history record corresponding to the match further comprises user commentary describing debugging operations performed during a previous stoppage of the code while in a state matching the current state (Fig 6, item 130).

Claim 34: Bates disclose a computer system for debugging code as in claim 30 above, and further discloses the debugger is configured to display a user interface configured to allow the user to create a history record for the at least one evaluation ([0042-0047], Fig 6).

Claim 35: Bates discloses a computer system for debugging code as in claim 34 above, and further disclose the user interface is configured to allow the user to specify a state to be associated with the history record to be created ([0047]).

Claim 36: Bates discloses a computer system for debugging code as in claim 34 above, and further discloses the at least one evaluation comprises a plurality of evaluations and wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select (menu selection, tool bar button, dialog box) in order to designate one of the plurality of evaluations as a primary state for the history record to be created, wherein the primary state is used to determine a match between a state described by code state information contained in the history record to be created and a subsequent state at a subsequent halted position in the code ([0040; 0047]).

Claim 37: Bates discloses a computer system for debugging code as in claim 34 above, and further discloses the at least one evaluation comprises a

plurality of evaluations and wherein the user interface displays the plurality of evaluations as selectable items from which the user is allowed to select in order to designate which of the plurality of evaluations are to be included in the history record to be created (Fig 6, item 130).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claim 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al (US 6,077,312).

9. Claim 19: Bates discloses a computer readable medium containing a debugger program which, when executed, performs an operation to facilitate debugging of code as in claim 18 above, but does not explicitly discloses the debugger program allows a user to enter a new record in the debug history repository if a matching record is not found. However, it would be obvious to one having ordinary skill in the art at the time the invention was made that to enter a new record in the debug history repository if a matching record is not found. This is the same as adding a new record to the debug history repository (Fig 6, item

160). The techniques are the same, adding records. Therefore, it would have been obvious to enter a new record in the debug history repository if a matching record is not found. One would have been motivated to enter a new record in the debug history repository if a matching record is not found in order to maintain the history records.

Claim 19: Bates discloses a computer readable medium containing a debugger program which, when executed, performs an operation to facilitate debugging of code as in claim 18 above, but does not explicitly discloses the debugger program automatically creates a new record in the debug history repository if a matching record is not found. However, it would be obvious to one having ordinary skill in the art at the time the invention was made that to have a program automatically creates a new record in the debug history repository if a matching record is not found. This is the same as automatically adding a new record to the debug history repository (Fig 3, item 62). The techniques are the same, adding records. Therefore, it would have been obvious to automically enter a new record in the debug history repository if a matching record is not found. One would have been motivated to automatically enter a new record in the debug history repository if a matching record is not found in order to maintain the history records.

CONCLUSION

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Goli et al (US 6,418,543 B1) discloses apparatus and method for debugging source code.

b. Takuma et al (US 6,141,791) discloses debug aid device, program compiler device, storage medium storing computer readable debugger program, and storage medium storing program compiler program.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Friday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, James Myhre can be reached on (571) 270-1065. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
PN
6/23/2006

James W. Myhre
Supervisory Primary Examiner